Claims

[c1] DE920000099US1

1.A method for operating an out of order processor in which a rename process is comprised of the pipeline an instruction stream is processed with, the method comprising the steps of:

processing the pipeline in a compressed way thereby risking dependency conflicts:

providing a separate logic for detecting a dependency conflict associated with an instruction currently to be renamed; setting a conflict flag reflecting the detection result; and

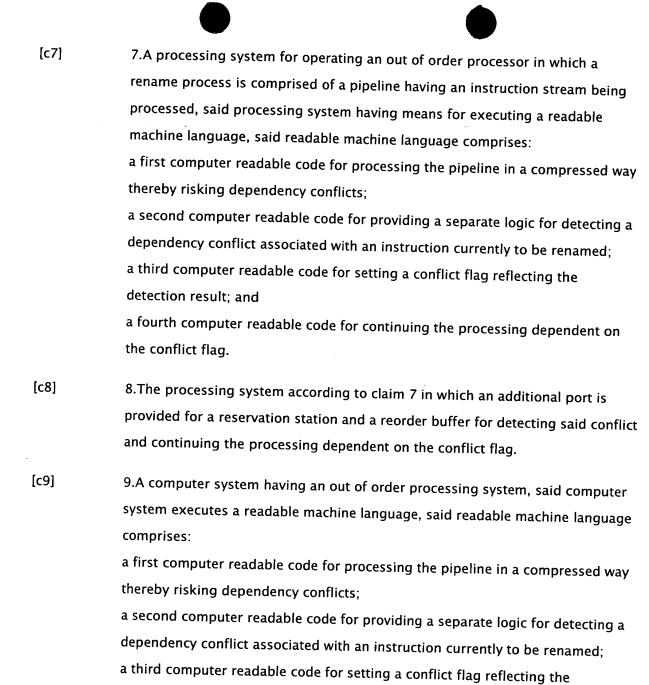
continuing the processing dependent on the conflict flag.

- 2. The method according to claim 1 in which the step of continuing the processing comprises the step of setting an interrupt bit in order to reset the pipeline.
- 3. The method according to claim 1 in which the step of continuing the processing comprises the step of flushing the pipeline.
 - 4. The method according to claim 1 in which the step of continuing the processing comprises the step to provide a reservation station (RS) with the missing information which caused the conflict, and continuing to process the same pipeline status without resetting or flushing it.
- [c5] 5. The method according to claim 4 in which the step of detecting said dependency conflict comprises the steps of: reading an instruction tag and a valid bit of a ReOrder Buffer (ROB) entry; determining that said valid bit was modified without being tracked by the RS; setting said conflict bit for indicating that said entry has to be issued to the RS; issuing said tag to an additional port of the tag compare logic; and triggering the write of said valid bit into a respective field of said RS.
- [c6] 6. The method according to claim 5 in which the ROB entry comprises result data currently written by an Instruction Execution Unit (IEU), further comprising the step of copying said result data into the respective entry of the RS.

[c2]

[c3]

[c4]



a fourth computer readable code for continuing the processing dependent on

detection result; and

the conflict flag.